Learning the Morphological Diversity

Gabriel Peyré, Jalal M. Fadili, Jean-Luc Starck

▶ **To cite this version:**

Gabriel Peyré, Jalal M. Fadili, Jean-Luc Starck. Learning the Morphological Diversity. SIAM Journal on Imaging Sciences, 2010, 3 (3), pp.646-669. 10.1137/090770783 . hal-00415782v2

HAL Id: hal-00415782
https://hal.science/hal-00415782v2

Submitted on 7 Jul 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

LEARNING THE MORPHOLOGICAL DIVERSITY

GABRIEL PEYRÉ*, JALAL FADILI†, AND JEAN-LUC STARCK‡

Abstract. This article proposes a new method for image separation into a linear combination of morphological components. Sparsity in fixed dictionaries is used to extract the cartoon and oscillating content of the image. Complicated texture patterns are extracted by learning adapted local dictionaries that sparsify patches in the image. These fixed and learned sparsity priors define a non-convex energy and the separation is obtained as a stationary point of this energy. This variational optimization is extended to solve more general inverse problems such as inpainting. A new adaptive morphological component analysis algorithm is derived to find a stationary point of the energy. Using adapted dictionaries learned from data allows to circumvent some difficulties faced by fixed dictionaries. Numerical results demonstrate that this adaptivity is indeed crucial to capture complex texture patterns.

Key words. Adaptive morphological component analysis, sparsity, image separation, inpainting, dictionary learning, cartoon images, texture, wavelets.

AMS subject classifications. 41A25, 42C40, 65T60

Morphological diversity is a concept where an image is modeled as a sum of components, each of these components having a given morphological signature. Sparsity in a redundant dictionary can be used to discriminate between these signatures, and fast algorithms such as the Morphological Component Analysis (MCA) have been developed in order to reconstruct simultaneously all the morphological components [54]. For natural images, containing edges and contours, two fixed dictionaries can be used such as the local DCT for sparsely representing the texture and the curvelet for the edges [55]. Results were interesting, but this approach bears some limitations since complicated textures may not be well represented by the local DCT, leading to a bad separation.

This paper extends the morphological diversity by learning the morphologies of complicated texture layers to help the separation process. These learned dictionaries are coupled with more traditional fixed morphologies to characterize the cartoon and oscillating content of an image. A new adaptive morphological component analysis algorithm performs iteratively both the learning and the separation.

1. Image Separation and Inverse Problem Regularization.

1.1. Image Separation. The image separation process decomposes an input image $u \in \mathbb{R}^N$ of N pixels into a linear combination of $|\Lambda|$ layers $\{u_s\}_{s\in\Lambda}$, the so-called morphological components

$$u = \sum_{s\in\Lambda} u_s + \varepsilon, \tag{1.1}$$

where ε is an error term representing the noise and model imperfections. Each u_s accounts for a different kind of features of the original data u.

*Ceremade, Université Paris-Dauphine, Place du Maréchal De Lattre De Tassigny, 75775 Paris Cedex 16, (FRANCE) (gabriel.peyre@ceremade.dauphine.fr)

†GREYC CNRS-ENSICAEN-Université de Caen 6, Bd du Maréchal Juin, 14050 Caen Cedex (FRANCE) (jalal.fadili@greyc.ensicaen.fr)

‡Service d'Astrophysique, CEA/Saclay, Orme des Merisiers, Bat 709, 91191 Gif-sur-Yvette Cedex (FRANCE) (jstarck@cea.fr)

Image separation is usually achieved by solving a variational optimization problem of the form

$$\min_{\{u_s\}_{s\in\Lambda}} \frac{1}{2}\Big\|u - \sum_{s\in\Lambda} u_s\Big\|^2 + \mu \sum_{s\in\Lambda} E_s(u_s), \tag{1.2}$$

where each energy $E_s : \mathbb{R}^N \mapsto \mathbb{R}^+$ favors images with some specific kind of structures. More precisely, for successful separation, each energy E_s is designed to be as small as possible over the layer u_s it is serving, while being large (or at least not as small) over the other components. Thus each of these layers has its attached prior E_s, and multiplying the number of priors might help to recover intricate image structures such as smooth areas, edges and textures of natural images.

1.2. Inverse Problems Regularization. Many problems in image processing can be cast as inverting a linear system $f = \mathcal{K}u + \varepsilon$ where $u \in \mathbb{R}^N$ is the data to recover, $f \in \mathbb{R}^m$ is the observed image and ε is a gaussian white noise of known finite variance. The bounded linear operator $\mathcal{K} : \mathbb{R}^N \mapsto \mathbb{R}^m$ is typically ill-behaved since it models an acquisition process that entails loss of information. This yields ill-posedness of the inverse problem.

This inversion problem is regularized by adding some prior knowledge on the typical structures of the original image u. This prior information accounts for the smoothness of the solution and can range from uniform smoothness assumption to more complex knowledge of the geometrical structures of u. The decomposition problem (1.2) is extended to handle an ill-behaved operator $\mathcal{K}$ which yields the following minimization problem

$$\min_{\{u_s\}_{s\in\Lambda}} \frac{1}{2}\Big\|f - \mathcal{K}\sum_{s\in\Lambda} u_s\Big\|^2 + \mu \sum_{s\in\Lambda} E_s(u_s)\ . \tag{1.3}$$

The simple case of image separation (1.1) corresponds to $\mathcal{K} = \mathrm{Id}_N$. Typical examples of inverse problems include deconvolution, inpainting and super-resolution. In the latter, one seeks to recover a high-resolution image u from a low-resolution observation f. In such a case, $\mathcal{K}$ is the convolution by a blurring kernel followed by a sub-sampling, and f lacks the high frequency content of u.

There is a flurry of research activity on linear inverse problems regularization in image processing. Comprehensive overviews can be found in dedicated monographs.

Sparsity-based regularization (e.g. in the wavelet domain) methods have recently received considerable attention, either by adopting a Bayesian expectation-maximization framework [26, 27, 6], by introducing surrogate functionals [14], or using a proximal forward-backward splitting framework [12, 24]. This framework has been successfully applied to inpainting [22, 24], deconvolution [25], multichannel data restoration and blind source separation [61, 7].

1.3. Image Inpainting. This paper considers the inpainting inverse problem, although our methods applies to other linear inverse problems as well. Inpainting is to restore missing image information based upon the still available (observed) cues from destroyed or deliberately masked subregions of the image f.

The inpainting problem corresponds to a diagonal operator

$$\mathcal{K} = \mathrm{diag}_i(\eta_i) \qquad \text{where} \qquad \eta_i = \begin{cases} 1 & \text{if} \quad i \notin \Omega \\ 0 & \text{if} \quad i \in \Omega \end{cases},$$

where $\Omega \subset \{0, \ldots, N-1\}$ denotes the set of missing pixels.

Inpainting of non-textured images has been traditionally approached by diffusion equations that progressively fills the missing pixels. The original work of Masnou and Morel makes use of the continuation law of the level sets [40]. Following their work, several authors proposed high order PDEs, see for instance [10, 5, 3] and anisotropic diffusion [57] for non-texture inpainting. The inpainting of complicated textures can be achieved by using copy-and-paste methods from computer graphics [19, 13] and can successfully inpaint large missing regions. The Morphological Component Analysis (MCA) is able to solve the inpainting problem [22, 24] for images containing simple textural content such as locally parallel oscillations. Methods based on learning sparse dictionaries [36, 33] and training fields of experts [51] are able to inpaint small missing regions.

2. Morphological Diversity Modeling with Sparsity.

2.1. Cartoon Modeling. For the sketchy part of an image, a usual prior is to assume that it belongs to some non-linear Banach space that favors the discontinuities in the image. In particular, this entails that the functional norm of the sketchy part in such spaces is small. Such spaces include the bounded variation (BV) space with the associated total variation norm introduced by Rudin, Osher and Fatemi [52]. Another important prior exploits the sparsity of wavelet coefficients, which corresponds to various kinds of Besov norms. A standard example of such sparsity-promoting prior is the ℓ^1-norm introduced by Donoho and Johnstone [17] in the wavelet context for denoising purposes.

Wavelets are however sub-optimal to efficiently capture edge singularities distributed along smooth curves. The curvelet tight frame, introduced by Candès and Donoho [9], is able to better represent cartoon images with smooth edges. Sparsity in a curvelet tight frame can thus improve the modeling of edges.

2.2. Oscillating Texture Modeling. Simple texture models can also be defined through variational energies that favor oscillations in images. Toward this goal, Meyer introduced a functional space where oscillating patterns have a small norm. It turns out that this Banach space is close to the dual of the BV space [42]. Meyer has defined the so-called G-norm that can be used to perform the decomposition of an image into a cartoon component (using for instance the total variation seminorm) and an oscillating texture component.

The work of Meyer paved the way to an active research area in variational image processing: cartoon+texture image decomposition. Several authors have proposed algorithms to solve Meyer's problem or close extensions, such as [60, 2, 32] to cite only a few.

Sparsity-based energies have also been proposed to decompose an image into cartoon+oscillating texture. To this end, Starck and co-authors [55, 54, 22] introduced the Morphological Component Analysis (MCA) framework, where overcomplete fixed dictionaries (one for each layer) are used as a source of diversity to discriminate between the components. The key is that each dictionary must sparsify the corresponding layer while being highly inefficient in representing the other content. For example, MCA is capable of decomposing an image into structure+oscillating texture, using the wavelet or curvelet dictionary for the cartoon layer, and the frame of local cosines for the oscillating texture.

Other dictionaries can enhance over the results of local cosines to capture warped locally oscillatory patterns. For instance, the waveatoms of Demanet and Ying [15]

and the brushlets of Meyer and Coifman [41] have been designed for this goal.

However, the standard MCA is intrinsically limited by the discriminative performance of its fixed non-adaptive dictionaries. Obviously, the latter are not able to sparsify complex textures appearing in natural images.

2.3. Adaptivity and Dictionary Learning. To enhance the modeling of complicated edge layouts and intricate texture patterns, one needs to resort to adapted energies, that are tuned to fit the geometry of complex images.

A class of adaptive methods consists in using a family of orthogonal bases and look for the best basis in this family using combinatorial optimization algorithms. The wedgelets [16] and the bandlets [29, 39] better represent contours than a traditional wavelet dictionary. For oscillating textures, a proper basis of the wavelet packet tree [37] with an appropriate tiling of the frequency domain sparsifies some oscillatory patterns. Cosine packets allow a dyadic partition of the spatial domain [37] according to a quad-tree structure. Grouplet bases [38] are able to approximate efficiently oscillating and turbulent textures, and were successfully applied to texture synthesis and inpainting in [48].

In contrast to these approaches, which are able to handle only a particular kind of images or textures, other approaches can adapt to the content of images through a learning process. By minimizing a sparsity criterion, such algorithms allow to optimize a local dictionary for a set of exemplar patches.

Olshausen and Field [45] were the first to propose a way of learning the dictionary from the data and to insist on the dictionary redundancy. They have applied this learning scheme to patches extracted from natural images. The major conclusion of this line of research is that learning over a large set of disparate natural images leads to localized oriented edge filters. Since then, other approaches to sparse coding have been proposed using independent components analysis [4], or different sparsity priors on the representation coefficients [31, 28, 23, 1].

It is worth point out that this dictionary learning bears tight similarities with sparsity-based blind source separation (BSS) algorithms as proposed in [61] and in the GMCA algorithm [7]. It is also related to non-negative matrix factorization that is used for source separation, see for instance [47]. The role played by the dictionary parallels the one of the mixing matrix in BSS.

These learned dictionaries have proven useful to perform image denoising [20], inpainting [36, 33], texture synthesis [49] and image recognition [35].

A preliminary description of the adaptive MCA method was presented in [50]. Shoham and Elad have developed in [53] an adaptive separation method that approximately minimizes our adaptive MCA energy, that is faster if no global dictionary is used.

2.4. Contributions. This paper proposes a new adaptive image separation method. It extends previous work on morphological component analysis by adapting the dictionary used to model and discriminate complex texture layer(s). Section 3 introduces the notions of fixed and learned dictionaries, that can be combined to achieve high quality separation. The local dictionaries are learned during the separation within a new adaptive morphological component analysis algorithm detailed in Section 4. This algorithm converges to a stationary point of a non-convex variational energy. Numerical results show that a single adaptive texture layer can be trained without user intervention to extract a complicated texture. Additional layers can be added as well, and in this case, a proper user-defined initialization is required to drive the algorithm to the desired decomposition which corresponds to a particular

stationary point of the energy. This option offers some flexibility to the user to guide the decomposition algorithm toward the desired solution.

3. Fixed and Adaptive Sparsity Energies. Each energy $E_s(u_s)$ depends on a dictionary D_s that is a collection of atoms used to describe the component u_s sparsely. This paper considers both fixed dictionaries, that are used to describe the content of the component as a whole, and local learned dictionaries that are applied to patches extracted from the component.

The set of indexes is thus decomposed as $\Lambda = \Lambda^{\mathrm{F}} \cup \Lambda^{\mathrm{L}}$, and a fixed layer $s \in \Lambda^{\mathrm{F}}$ is assigned an energy $E^{\mathrm{F}}(u_s, D_s)$, whereas an energy $E^{\mathrm{L}}(u_s, D_s)$ is attached to a layer $s \in \Lambda^{\mathrm{L}}$ with a learned dictionary. The fixed dictionaries are defined by the user and correspond typically to a cartoon morphological component or simple oscillating patterns. On the contrary, learned dictionaries $\{D_s\}_{s\in\Lambda^{\mathrm{L}}}$ are optimized by our algorithm to capture complicated stationary texture patterns. The corresponding adaptive separation process thus extends (1.2) to an optimization on both the components and the adaptive dictionaries

$$\min_{\{u_s\}_{s\in\Lambda},\{D_s\in\mathcal{D}_s\}_{s\in\Lambda^{\mathrm{L}}}} \frac{1}{2}\Big\|f-\mathcal{K}\sum_{s\in\Lambda}u_s\Big\|^2+\mu\sum_{s\in\Lambda^{\mathrm{F}}}E^{\mathrm{F}}(u_s,D_s)+\mu\sum_{s\in\Lambda^{\mathrm{L}}}E^{\mathrm{L}}(u_s,D_s), \quad (3.1)$$

where $\mathcal{D}_s$ is a suitable set of convex constraints to be defined later (see (4.3)).

3.1. Sparsity-based Energy for a Fixed Dictionary. A fixed dictionary $D_s = (d_{s,j})_{0\leqslant j<m_s}$, for $s \in \Lambda^{\mathrm{F}}$, is a (possibly redundant) collection of $m_s \geqslant N$ atoms $d_{s,j} \in \mathbb{R}^N$, that can be represented as a rectangular matrix $D_s \in \mathbb{R}^{N\times m_s}$. The decomposition of a component u_s using this dictionary reads

$$u_s = D_s x_s = \sum_{j=0}^{m_s-1} x_s[j] d_{s,j}.$$

For a redundant dictionary where $m_s > N$, such a decomposition is non-unique, and a sparsity-promoting energy favors sparse coefficients x_s, for which most of the entries $x_s[j]$ are zero. In this paper, we use a convex ℓ^1 sparsity measure

$$\|x_s\|_1 = \sum_j |x_s[j]|,$$

which was proposed by Chen, Donoho and Saunders [11] in the popular basis pursuit denoising problem (BPDN or Lasso for statisticians) for sparse approximation.

Finding a sparse approximation $D_s x_s$ of u_s in D_s can then be formulated as minimizing the following energy

$$E^{\mathrm{F}}(u_s, D_s) = \min_{x_s\in\mathbb{R}^{m_s}} \mathcal{E}^{\mathrm{F}}(u_s, x_s, D_s) \ , \quad (3.2)$$

$$\text{where} \quad \mathcal{E}^{\mathrm{F}}(u_s, x_s, D_s) = \frac{1}{2}\|u_s - D_s x_s\|^2 + \lambda\|x_s\|_1. \quad (3.3)$$

The parameter λ allows an approximate sparse representation $D_s x_s \approx u_s$ and should be adapted to the level of the noise and the sparsity of the sources.

Cartoon sparse models. Wavelets [37] are used extensively in image compression and allow to capture efficiently images with isotropic singularities and images with bounded variations. We use a redundant dictionary D_{wav} of translation invariant wavelets to capture the sketchy content of an image, which is assumed to have a small total variation.

To capture more regular edge patterns, we use a redundant tight frame of curvelets D_{curv}, introduced by Candès and Donoho [9] to represent optimally cartoon images with C^2-regular edge curves.

Oscillating sparse models. Locally oscillating and stationary textures are handled with a redundant tight frame D_{dct} of local cosines [37]. We use local cosine atoms defined on blocks of 32×32 pixels, that are zero outside the blocks and are thus vectors of length N. We use an overlapping factor of 2 of the blocks along the horizontal and vertical directions, so that the redundancy of D_{dct} is $m_s/N = 4$. As explained in the introduction, other dictionaries well-suited for sparsifying oscillating patterns could be used as well, e.g. waveatoms [15].

3.2. Sparsity-based Energy for a Learned Dictionary. We use local learned dictionaries $\{D_s\}_{s\in\Lambda^{\mathrm{L}}}$ to capture fine scale structures of complex textures. For $s \in \Lambda^{\mathrm{L}}$, a local learned dictionary $D_s \in \mathbb{R}^{n\times m_s}$ is used to represent patches $R_k(u_s) \in \mathbb{R}^n$ of $n = \tau \times \tau$ pixels extracted from a component u_s,

$$\forall\, 0 \leqslant k_{1,2} < \sqrt{N}/\Delta \quad \text{and} \quad -\tau/2 \leqslant i_{1,2} < \tau/2, \quad R_k(u_s)[i] = u_s(k_1\Delta + i_1, k_2\Delta + i_2),$$

where $i = (i_1, i_2)$ is the location of a pixel in the patch, $k = (k_1, k_2)$ indexes the central location of the patch, and $1 \leqslant \Delta \leqslant \tau$ controls the sub-sampling of the patch extraction process. Although k is a pair of integers to index 2D patches, it is conveniently converted into a single integer in $\{0, \ldots, N/\Delta^2 - 1\}$ after rearranging the patches in a vectorized form to store them as column vectors in a matrix.

Similarly to the energy (3.2) associated to a fixed dictionary, we define an energy $E^{\mathrm{L}}(u_s, D_s)$ associated to a local dictionary D_s. This energy allows one to control the sparsity of the decomposition of all the patches $R_k(u_s)$ in D_s. Following Aharon and Elad [20, 1], we define this energy $E^{\mathrm{L}}(u_s, D_s)$ as

$$E^{\mathrm{L}}(u_s, D_s) = \min_{\{x_{s,k}\}_k \in \mathbb{R}^{m_s \times N/\Delta^2}} \mathcal{E}^{\mathrm{L}}(u_s, \{x_{s,k}\}_k, D_s), \tag{3.4}$$

$$\text{where} \quad \mathcal{E}^{\mathrm{L}}(u_s, \{x_{s,k}\}_k, D_s) = \frac{1}{p}\sum_k \left(\frac{1}{2}\|R_k(u_s) - D_s x_{s,k}\|^2 + \lambda\|x_{s,k}\|_1\right), \tag{3.5}$$

where $p = (\tau/\Delta)^2 = n/\Delta^2$. Each $x_{s,k}$ corresponds to the coefficients of the decomposition of the patch $R_k(u_s)$ in the dictionary D_s. The weight $1/p$ in the energy (3.4) compensates for the redundancy factor introduced by the overlap between the patches $R_k(u_s)$. This normalization allows one to re-scale the learned dictionary energy (3.4) to be comparable with the fixed dictionary one (3.2).

3.3. Images vs. Coefficients. The variational formulation (1.3) proposed in this paper directly seeks for the components $\{u_s\}_{s\in\Lambda}$, and the coefficients are only considered as auxiliary variables. Alternative formulations of inverse problems in redundant dictionaries or union of bases would look instead for the coefficients x_s or $\{x_{s,k}\}_k$ of each component in the dictionary D_s.

The corresponding coefficient-based minimization reads

$$(\{x_s^\star\}_{s\in\Lambda}, \{D_s^\star\}_{s\in\Lambda^{\mathrm{L}}}) \in \underset{\{x_s\}_{s\in\Lambda}, \{D_s\in\mathcal{D}_s\}_{s\in\Lambda^{\mathrm{L}}}}{\operatorname{argmin}} \tag{3.6}$$

$$\frac{1}{2}\Big\| f - \mathcal{K}\sum_{s\in\Lambda^{\mathrm{F}}} D_s x_s - \mathcal{K}\sum_{s\in\Lambda^{\mathrm{L}},k} R_k^*(D_s x_{s,k})\Big\|^2 + \lambda\sum_{s\in\Lambda^{\mathrm{F}}}\|x_s\|_1 + \lambda\sum_{s\in\Lambda^{\mathrm{L}},k}\|x_{s,k}\|_1, \tag{3.7}$$

where the dual operator R_k^* reconstruct an image in $\mathbb{R}^N$ with zero values outside the patch. Such a coefficient-based optimization is used for fixed dictionaries in [55, 22, 24].

A fixed dictionary component for $s \in \Lambda^{\mathrm{F}}$ is retrieved from these optimized coefficients as $u_s^\star = D_s x_s^\star$. For a local learned dictionary, the reconstruction of the patches are summed

$$\forall\, s \in \Lambda^{\mathrm{L}}, \quad u_s^\star = \sum_k R_k^*(D_s^\star x_{s,k}^\star). \tag{3.8}$$

The reconstruction formula (3.8) shows that the optimization (3.7) for the learned component $s \in \Lambda^{\mathrm{L}}$ corresponds to finding a sparse approximation of u_s in the highly redundant dictionary $\{R_k^*(d_{s,j})\}_{k,j}$ that gathers all the atoms at all patch locations.

The two formulations (1.3) and (3.7) are expected to differ significantly. These differences share some similarities with the formulations analyzed in [21], that studies analysis and synthesis signal priors. In our setting, where we use local learned dictionaries, the formulation (1.3) over the image domain makes more sense. In this formulation, each patch is analyzed independently by the sparsity prior, and the ℓ^2 fidelity term gathers linearly the contributions of the patches to obtain u_s. As noticed by Aharon and Elad [20, 1], this average of sparse representations has some flavor of minimum mean square estimation, which further helps to reduce the noise.

Furthermore, the formulation (3.7) corresponds to the optimization of coefficients in a highly redundant dictionary, which is demanding numerically. In contrast, our formulation (1.3) allows for an iterative scheme that optimizes the coefficients independently over each patch and average them afterward. We describe this adaptive MCA scheme in the following section.

The last chief advantage of (1.3) is that it decouples the contribution of each local learned dictionary D_s, for $s \in \Lambda^{\mathrm{L}}$. This simplifies the learning process, since each dictionary is independently optimized during the iterations of our adaptive MCA.

4. Adaptive Morphological Component Analysis. The morphological component analysis (MCA) algorithm [55, 54] allows to solve iteratively the variational separation problem (1.3) for sparsity-based energies E_s as defined in (3.2). For the decomposition of an image into its geometrical and textured parts, the original approach [55, 54] uses fixed dictionaries of wavelets D_{wav}, curvelets D_{curv}, and local cosines D_{dct}. This paper extends the MCA algorithm to deal with energies E_s associated to local learned dictionaries D_s as defined in (3.4). In addition, our adaptive MCA algorithm is able to optimize the local dictionaries D_s, which are automatically adapted to the texture to extract.

4.1. Adaptive Variational Problem. The new adaptive MCA algorithm minimizes iteratively the energy (1.3) by adding to the decomposition variables $\{u_s\}_{s\in\Lambda}$ and $\{D_s\}_{s\in\Lambda^{\mathrm{L}}}$ auxiliary variables $\{x_s\}_{s\in\Lambda}$ corresponding to the coefficients of the decomposition of each u_s. For a fixed dictionary layer $s \in \Lambda^{\mathrm{F}}$, these coefficients are

stored in a vector $x_s \in \mathbb{R}^{m_s}$. For a local learned dictionary $s \in \Lambda^{\mathrm{L}}$ these coefficients are a collection of vectors $\{x_{s,k}\}_{k=0}^{N/\Delta^2-1} \in \mathbb{R}^{m_s \times N/\Delta^2}$.

The energy minimized by the adaptive MCA algorithm is

$$\mathcal{E}(\{u_s\}_s, \{x_s\}_s, \{D_s \in \mathcal{D}_s\}_{s\in\Lambda^{\mathrm{L}}}) = \frac{1}{2}\|f - \mathcal{K}\sum_s u_s\|^2 + \tag{4.1}$$

$$\mu \sum_{s\in\Lambda^{\mathrm{F}}} \mathcal{E}^{\mathrm{F}}(u_s, x_s, D_s) + \mu \sum_{s\in\Lambda^{\mathrm{L}}} \mathcal{E}^{\mathrm{L}}(u_s, x_s, D_s)\ , \tag{4.2}$$

where the fixed and learned energies $\mathcal{E}^{\mathrm{F}}$ and $\mathcal{E}^{\mathrm{L}}$ are defined in (3.3) and (3.5).

The constraint $D_s = (d_{j,s})_{0\leqslant j<m_s} \in \mathcal{D}_s$ for $s \in \Lambda^{\mathrm{L}}$ ensures that the columns $d_{i,s}$ of D_s have bounded norm

$$\forall\, j = 0, \ldots, m_s - 1,\ \|d_{j,s}\|^2 = \sum_i |d_{j,s}[i]|^2 \leqslant 1.$$

This avoids the classical scale indeterminacy between the dictionaries and the coefficients since replacing (D_s, x_s) by $(aD_s, x_s/a)$ for $a > 1$ leaves the data fidelity term unchanged, while diminishing the ℓ^1 penalty. We also impose that the atoms have zero mean, which is consistent with the intuition that a texture contains locally only high frequencies. The convex set $\mathcal{D}_s$ reads

$$\mathcal{D}_s = \left\{ D \in \mathbb{R}^{n\times m_s} \setminus \forall\, j = 0, \ldots, m_s - 1,\ \|d_{j,s}\| \leqslant 1 \text{ and } \sum_i d_{j,s}[i] = 0 \right\}. \tag{4.3}$$

Additional constraints. We note that other constraints could be used to better control the stability of the learned dictionaries. For instance, an orthogonality constraint has been considered by Lesage et al. [30]. It could also be incorporated in our constraint set $\mathcal{D}_s$. Additional constraints could be considered to improve the separation. In particular, enforcing the atoms of two distinct dictionaries D_s and $D_{s'}$ to have different morphologies might increase the quality of the separation. Investigation of such additional constraints is left for a future work.

Adaptive non-convex minimization. The energy $\mathcal{E}$ is marginally convex in each of its arguments, and is optimized over a convex set. However, $\mathcal{E}$ is non-convex jointly in all its arguments. We thus propose an iterative block relaxation coordinate descent minimization scheme, and show that it converges to a stationary point of the energy.

The adaptive MCA algorithm operates by minimizing successively and cyclically $\mathcal{E}$ on the set of components $\{u_s\}_{s\in\Lambda}$, on the set of coefficients $\{x_s\}_{s\in\Lambda}$ and the set of learned dictionaries $\{D_s\}_{s\in\Lambda^{\mathrm{L}}}$. Each minimization is performed while all remaining variables are held fixed.

The initialization of the dictionaries $\{D_s\}_{s\in\Lambda^{\mathrm{L}}}$ is thus important, and user intervention can improve the result by selecting initial features relevant for texture extraction.

4.2. Parameters Selection. Selecting optimal values for μ and λ is a delicate and difficult task. The parameter μ is adapted to the noise level. In the case where ε is a Gaussian white noise of variance σ, μ is set so that the residuals satisfy $\|f - \mathcal{K}\sum_s u_s^\star\| \approx \sqrt{N}\sigma$, which works well for separation and denoising applications. The same weight λ could be chosen for all dictionaries (i.e. components). Nevertheless, if additional prior knowledge on the amplitude of each component is available, it could be wisely incorporated by selecting different weights. In our experiments, we

set $\lambda = \sigma^2/30$, that was shown by Aharon and Elad [20, 1] to be a good choice for denoising applications.

The choice of the size of the learned dictionaries (size of the patch $n = \tau^2$ and redundancy m_s/n) is still largely an open question in dictionary learning. For inpainting, the size of the patches should be chosen larger than that of the missing regions. Increasing the redundancy of the dictionary enables to capture more complicated textures, but makes the learning more difficult both computationally (complexity increases with overcompleteness) and in terms of convergence behavior to a poor local minimum (conditioning of the dictionary deteriorates with increasing redundancy).

4.3. Step 1 – Update of the Coefficients $\{x_s\}_{s\in\Lambda}$. The update of the coefficients requires the minimization of $\mathcal{E}$ with respect to $\{x_s\}_{s\in\Lambda}$. Since this problem is separable in each of the coefficient vector x_s, we perform the optimization independently for the coefficients of each fixed layer or each patch in a learned dictionary layer.

For a fixed dictionary $s \in \Lambda^{\mathrm{F}}$, this corresponds to solving

$$x_s \in \operatorname*{argmin}_{x\in\mathbb{R}^{m_s}} \frac{1}{2}\|u_s - D_s x\|^2 + \lambda\|x\|_1. \tag{4.4}$$

For a learned dictionary $s \in \Lambda^{\mathrm{L}}$, the minimization is performed with respect to each patch index k

$$x_{s,k} \in \operatorname*{argmin}_{x\in\mathbb{R}^{m_s}} \frac{1}{2}\|R_k(u_s) - D_s x\|^2 + p\lambda\|x\|_1 \ . \tag{4.5}$$

Both (4.4) and (4.5) correspond to sparse coding by minimizing a basis pursuit denoising problem [11]. The literature is inundated by a variety of algorithms trying to solve this convex problem efficiently, among which interior point solvers [11], iterative soft thresholding [14, 12], or Nesterov multi-step scheme [43, 44].

4.4. Step 2 – Update of the Components $\{u_s\}_{s\in\Lambda}$. Updating the components $\{u_s\}_{s\in\Lambda}$ requires to solve a quadratic minimization problem

$$\min_{\{u_s\}_{s\in\Lambda}} \|f - \mathcal{K}\sum_{s\in\Lambda} u_s\|^2 + \mu \sum_{s\in\Lambda^{\mathrm{F}}} \|u_s - D_s x_s\|^2 + \frac{\mu}{p}\sum_{s\in\Lambda^{\mathrm{L}},k} \|R_k(u_s) - D_s x_{s,k}\|^2. \tag{4.6}$$

This is a high dimensional problem since it involves all the layers, and it can be solved with a conjugate gradient descent.

An alternate method consists in cycling repeatedly on each component u_s for $s \in \Lambda$, and optimizing (4.6) with respect to u_s alone. This generates iterates $u_s^{(\ell)}$ that ultimately converge to a global minimizer of (4.6). Although the convergence is slower than with a conjugate gradient descent, it is simpler to implement since each component update is easy to compute in closed-form.

At a step ℓ of this update of the components, a new iterate $u_s^{(\ell+1)}$ is obtained by minimizing (4.6) with respect to u_s alone. For a fixed dictionary $s \in \Lambda^{\mathrm{F}}$, this leads to

$$u_s^{(\ell+1)} = \operatorname*{argmin}_{u\in\mathbb{R}^N} \|r_s - \mathcal{K}u\|^2 + \mu\|u - D_s x_s\|^2 \quad \text{with} \quad r_s = f - \mathcal{K}\sum_{s'\neq s} u_{s'}^{(\ell)},$$

and for a learned dictionary $s \in \Lambda^{\mathrm{L}}$

$$u_s^{(\ell+1)} = \operatorname*{argmin}_{u\in\mathbb{R}^N} \|r_s - \mathcal{K}u\|^2 + \frac{\mu}{p}\sum_k \|R_k(u) - D_s x_{s,k}\|^2 \ .$$

Note that we use an equality and not an inclusion since the marginal objective is strictly (in fact strongly) convex in u, therefore having a unique global minimizer in this coordinate.

This leads to the following closed-form update rule

$$u_s^{(\ell+1)} = \left(\mathcal{K}^*\mathcal{K} + \mu \mathrm{Id}_N\right)^{-1} \left(\mathcal{K}^* r_s + \mu \hat{u}_s\right) \ , \tag{4.7}$$

where the reconstructed $\hat{u}_s$ is computed differently depending whether the dictionary is fixed or learned

$$\hat{u}_s = \begin{cases} D_s x_s, & \text{if} \quad s \in \Lambda^{\mathrm{F}}, \\ \frac{1}{p} \sum_k R_k^*(D_s x_{s,k}), & \text{if} \quad s \in \Lambda^{\mathrm{L}}. \end{cases} \tag{4.8}$$

To derive the expression for a learned dictionary, we used the fact that

$$\frac{1}{p} \sum_k R_k^* R_k = \mathrm{Id}_N,$$

and a special care should be taken at the boundaries of the image.

For a general operator $\mathcal{K}$, the update (4.7) requires to solve a well-conditioned linear system, which can be computed by conjugate gradient. If $\mathcal{K}$ is known to be diagonalized in some domain (e.g. Fourier for convolution), then (4.7) can be implemented very efficiently. For the image separation problem, where $\mathcal{K} = \mathrm{Id}_N$, the update of the component u_s reduces to the convex sum

$$u_s^{(\ell+1)} = (1+\mu)^{-1}(r_s + \mu \hat{u}_s).$$

For the inpainting problem, the update becomes

$$u_s^{(\ell+1)}[i] = \begin{cases} (1+\mu)^{-1}(r_s[i] + \mu \hat{u}_s[i]) & \text{if} \quad i \in \Omega, \\ \hat{u}_s[i] & \text{if} \quad i \notin \Omega. \end{cases}$$

4.5. Step 3 – Update of the Dictionaries $\{D_s\}_{s \in \Lambda^{\mathrm{L}}}$. This update step concerns only local learned dictionaries D_s for $s \in \Lambda^{\mathrm{L}}$. Since this problem is separable in each dictionary, this optimization is carried out independently for each D_s. This corresponds to

$$D_s \in \operatorname*{argmin}_{D \in \mathcal{D}_s} \sum_k \|R_k(u_s) - D x_{s,k}\|^2 = \operatorname*{argmin}_{D \in \mathcal{D}_s} \|U_s - D X_s\|_F^2 \tag{4.9}$$

where $\|.\|_F$ stands for the Frobenius norm, the convex set $\mathcal{D}_s$ is defined in (4.3), $U_s \in \mathbb{R}^{n \times N/\Delta^2}$ is the matrix whose kth column is $R_k(u_s) \in \mathbb{R}^n$, and $X_s \in \mathbb{R}^{m_s \times N/\Delta^2}$ is the matrix whose kth column is $x_{s,k} \in \mathbb{R}^{m_s}$.

The minimization of (4.9) is accomplished by means of a projected gradient descent, that computes iterates $D_s^{(\ell)}$ such that

$$D_s^{(\ell+1)} = \mathcal{P}_{\mathcal{D}_s}\left(D_s^{(\ell)} + \tau (U_s - D_s^{(\ell)} X_s) X_s^* \right) \ , \tag{4.10}$$

where $0 < \tau < 2/\|X_s X_s^*\|$ is the descent step-size, and

$$(\tilde{d}_j)_{j=0}^{m_s-1} = \mathcal{P}_{\mathcal{D}_s}(D)$$

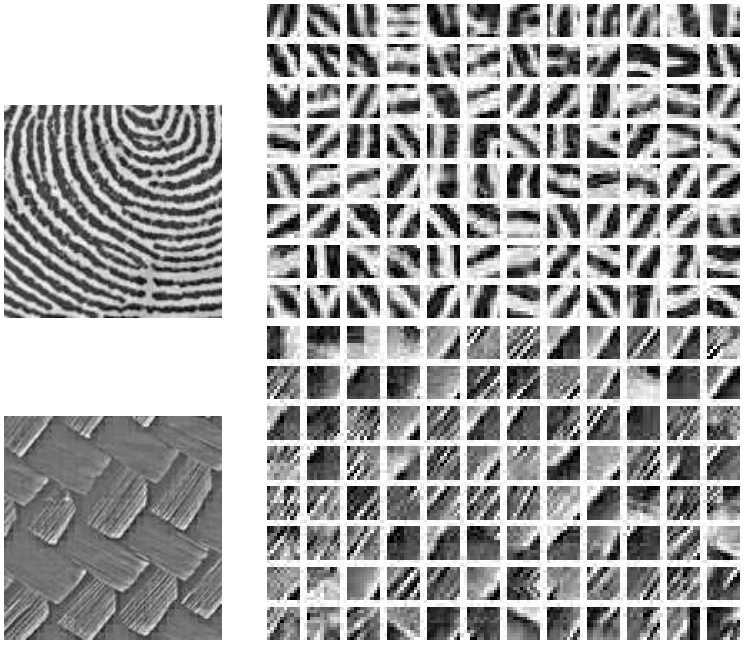
is the projection of $D = (d_j)_{j=0}^{m_s-1}$ on the convex set $\mathcal{D}_s$, that has an explicit expression

$$\tilde{d}_j = \frac{d_j - c}{\|d_j - c\|} \quad \text{with} \quad c = \frac{1}{n}\sum_{i=0}^{n-1} d_j[i].$$

We note that the linear constraint $\sum_i d_{j,s}[i] = 0$ can be dropped if each patch $R_k(u_s)$, as a column vector in the matrix U_s, is centered by subtracting its mean prior to the dictionary update.

It is worth noting that potentially more efficient schemes, such as Nesterov multi-step descent [43, 44] or block-coordinate minimization [33], could be used to solve (4.9). Other approximate methods, potentially faster, have been used to perform the update of the dictionary [23, 1], but they do not minimize (4.9) exactly.

Figure 4.1 shows two examples of dictionary learned from input exemplar textures. One can see that the learned atoms d_j do a good job at capturing the patterns of the exemplar texture.



Exemplars $\tilde{u}_1$ and $\tilde{u}_2$ Dictionaries D_1 and D_2

FIG. 4.1. *Examples of dictionary learning on two textures, with patches of size $\tau \times \tau = 10 \times 10$ pixels (only a subset of the atoms is displayed on the right).*

4.6. Adaptive MCA algorithm. The adaptive MCA scheme is summarized in Algorithm 1. It iteratively cycles between the above three steps. Each step is carried out using an inner iterative algorithm, with convergence tolerances η_{coef}, η_{comp} and η_{dico} associated to each of the corresponding inner iteration.

Initialization of the dictionaries. Since the energy $\mathcal{E}$ minimized by the adaptive MCA is non-convex, different initializations for the dictionaries $\{D_s\}_{s\in\Lambda^{\text{L}}}$ might lead to different solutions. In the numerical experiments detailed in Section 5, we consider several initialization scenarios, that may require some user intervention.

Algorithm 1: Adaptive Morphological Component Analysis.

Input: observation f, fixed dictionaries $\{D_s\}_{s\in\Lambda^{\mathrm{F}}}$, parameters μ and λ ;
Initialization: $\forall\, s\in\Lambda, u_s=0$, $\forall\, s\in\Lambda^{\mathrm{L}}$ initialize D_s ;
while not converged **do**
  begin *Update the coefficients:*
    for each $s\in\Lambda^{\mathrm{F}}$ **do**
      compute x_s by minimizing (4.4) with tolerance η_{coef},
    for each $s\in\Lambda^{\mathrm{L}}$, each k **do**
      compute $x_{s,k}$ by minimizing (4.5) with tolerance η_{coef}.
  end
  begin *Update the components:* set $\ell=0$, $\forall\, s\in\Lambda, u_s^{(0)}=u_s$
    repeat
      for each $s\in\Lambda$ **do**
        compute $u_s^{(\ell+1)}$ using (4.7).
    until $\max_s \|u_s^{(\ell+1)}-u_s^{(\ell)}\|<\eta_{\mathrm{comp}}$;
    Set $\forall\, s, u_s=u_s^{(\ell+1)}$;
  end
  begin *Update the dictionaries:*
    for each $s\in\Lambda^{\mathrm{L}}$ **do** set $\ell=0$, $D_s^{(0)}=D_s$
      repeat
        compute $D_s^{(\ell+1)}$ using (4.10).
      until $\|D_s^{(\ell+1)}-D_s^{(\ell)}\|<\eta_{\mathrm{dico}}$;
      Set $D_s=D_s^{(\ell+1)}$.
  end
Output: Estimated components $\{u_s\}_{s\in\Lambda}$.

Convergence of adaptive MCA. The following result ensures that this adaptive MCA algorithm converges to a stationary point of the energy $\mathcal{E}$.

PROPOSITION 4.1. *Suppose that each of steps 1-3 is solved exactly by the adaptive MCA algorithm. Then, the obtained sequence of iterates is defined, bounded and every accumulation point is a stationary point of* $\mathcal{E}$.

Proof. The optimization problem (4.1) has at least one solution by coercivity. The convergence proof of the block-relaxation minimization scheme follows, after identifying our problem with the one considered by the author in [59]. Indeed using a comparable notation to that of [59], we can write $\mathcal{E}(\{u_s\}_s,\{x_s\}_s,\{D_s\}_{s\in\Lambda^{\mathrm{L}}})$ as

$$J_0(\{D_s\}_s,\{u_s\}_s,\{x_s\}_s)+J_D(\{D_s\}_{s\in\Lambda^{\mathrm{L}}})+J_u(\{u_s\}_s)+\sum_s J_x^s(x_s)$$

where

$$J_0(\{D_s\}_s,\{u_s\}_s,\{x_s\}_s)=\frac{\mu}{2}\sum_{s\in\Lambda^{\mathrm{F}}}\|u_s-D_sx_s\|^2+\frac{\mu}{2p}\sum_{s\in\Lambda^{\mathrm{L}},k}\|R_k(u_s)-D_sx_{s,k}\|^2$$

$$J_D(\{D_s\}_s) = \begin{cases} 0 & \text{if} \quad D_s \in \mathcal{D}_s \ \forall s \in \Lambda^{\mathrm{L}}, \\ +\infty & \text{if} \quad D_s \notin \mathcal{D}_s \ \text{ for some } s \in \Lambda^{\mathrm{L}}. \end{cases} ,$$

$$J_u(\{u_s\}_s) = \frac{1}{2}\|f - \mathcal{K}\sum_s u_s\|^2 ,$$

$$J_x^s(x_s) = \begin{cases} \mu\lambda\|x_s\|_1, & \text{(fixed dictionary)}, \\ \mu\lambda/p\sum_k \|x_{s,k}\|_1, & \text{(learned dictionary)}. \end{cases} , \forall s \in \Lambda.$$

It is not difficult to see that J_0 has a non-empty open domain and is continuously differentiable on its domain. Thus J_0 satisfies Assumption A.1 in [59]. Moreover, $\mathcal{E}$ is continuous on its effective domain, with bounded level sets. $\mathcal{E}$ is also convex in $(u_1, \ldots, u_{|\Lambda|}, x_1, \ldots, x_{|\Lambda|})$, and J_D is convex. Thus, Lemma 3.1 and Theorem 4.1(b) of [59] imply that the sequence of iterates provided by the block coordinate descent MCA algorithm is defined, bounded and every accumulation point is a stationary point of $\mathcal{E}$. □

It is important to note that the convergence is only guaranteed for an exact adaptive MCA that performs an exact coordinate-wise minimization at each of the three steps. Little is known about the behavior of an approximate block coordinate descent, and the tolerances $\eta_{\mathrm{coef}}, \eta_{\mathrm{comp}}$ and η_{dico} should be decayed through the iterations of the MCA to ensure convergence. For the numerical experiments of Section 5, we nevertheless used fixed tolerances, and we always observed empirically the convergence of the algorithm.

Varying threshold. An important feature of the morphological component analysis is that the value of the parameter μ is decreased through iterations until it reaches its final value that is adapted to the noise level. This allows to speed up the convergence, and is reminiscent of strategies employed in continuation and path following methods for solving BPDN [18, 46]. More sophisticated threshold update schedules might be used, see for instance [8].

Computational complexity. The bulk of computation in step 1 of Algorithm 1 is invested in the application of the matrix D_s and its adjoint D_s^*. For fixed dictionaries corresponding to tight frames used in this paper, these matrices are never explicitly constructed. Rather, they are implemented as fast implicit analysis and synthesis operators. The complexity of these operators for the wavelet D_{wav}, local DCT D_{dct} and curvelets D_{curv} is $O(N\log(N))$ operations. For learned dictionaries, the matrices $D_s \in \mathbb{R}^{n\times m_s}$ are explicitly constructed, but their size is much smaller than that of fixed dictionaries.

The number of required iterations depends on the tolerance η_{coef} and on the algorithm used for the minimization. Nesterov multi-step descent [43, 44] enjoys a fast decay of the ℓ^1-regularized objective with $O(1/\ell^2)$ rate on the objective. This much faster than the the convergence rate of iterative soft thresholding which is only $O(1/\ell)$. In practice, Nesterov multi-step scheme performs well.

Each iteration of step 2 necessitates to reconstruct $\hat{u}_s$ for each s, which can be typically achieved in $O(N)$ or $O(N\log N)$ operations for a fixed dictionary, and $O(m_s nN/\Delta^2)$ operations for a local learned dictionary. Then the linear system (4.7) must be solved. For separation and inpainting problems, this step is fast and costs at most $O(N)$ for each s. The complexity of each iteration of the projected gradient descent of step 3 is similar to the complexity of step 2.

5. Numerical Examples. Throughout all the numerical examples, we use patches of width $\tau = 10$ pixels for the local learned dictionaries, with an overlap $\Delta = \tau/2$.

For all experiments, we artificially add a Gaussian white noise ε of standard deviation $\sigma/\|f\|_\infty = 0.03$. The parameters λ and μ have been selected as explained in Section 4.2.

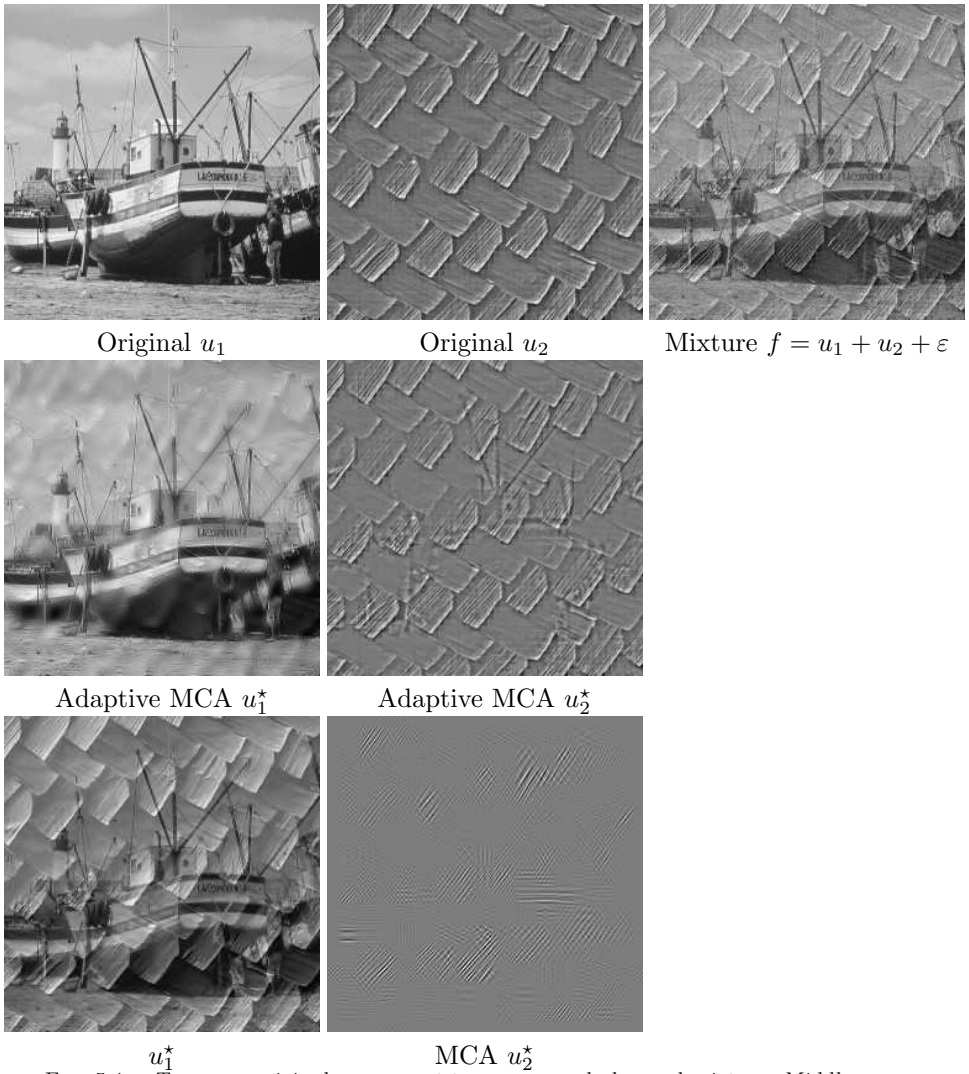


FIG. 5.1. *Top row: original component to recover and observed mixture. Middle row: separation using adaptive MCA with a wavelet dictionary and a learned dictionary. Bottom row: separation using MCA with a wavelet and a local DCT dictionary.*

5.1. Image Decomposition. We recall that the image decomposition problem corresponds to $\mathcal{K} = \mathrm{Id}_N$ in (4.1). One thus looks for an approximate decomposition $f \approx \sum_s u_s$.

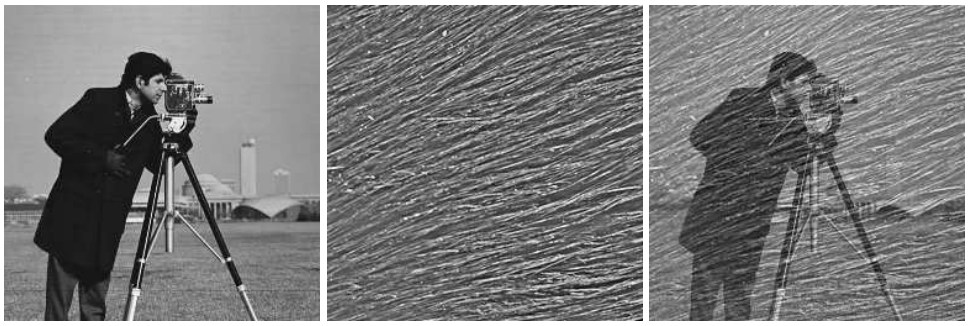
Image decomposition with a single adapted layer. We perform two experiments to study the behavior of our algorithm with a fixed dictionary D_1 to capture the sketchy (cartoon) part of the image, and an adapted dictionary D_2 to be learned in order to capture an additional homogeneous texture. When only $|\Lambda^{\mathrm{L}}| = 1$ adapted layer is computed, we found that the obtained results depend only slightly on the

initialization. In this case, D_2 is initialized with random patches extracted from the observations f.

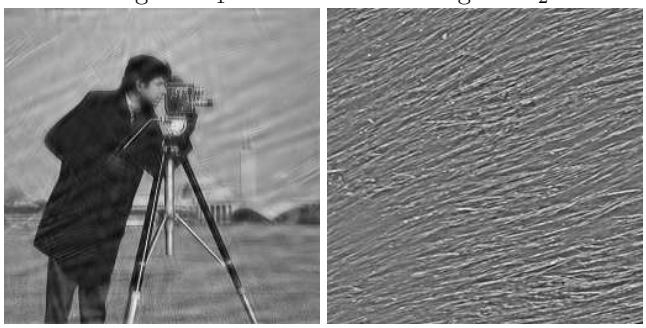
Figure 5.1, second row, shows the results of a first separation experiment where D_1 is a fixed redundant wavelet tight frame and D_2 is a learned dictionary. The adaptive layer is able to capture the fine scale details of the texture. Figure 5.1, third row, shows the results where D_2 is a fixed local DCT tight frame. This clearly demonstrates that the local DCT is not able to capture efficiently the details of the texture, and shows the usefulness and enhancement brought by adaptivity to the separation process.

Figure 5.2, first row, displays an example of separation where the layer u_1 has a strong cartoon morphology. We thus use a fixed curvelet dictionary $D_1 = D_{\text{curv}}$. The second layer corresponds to a learned dictionary. Figure 5.2, second row, shows the separation obtained with our adaptive MCA. Although the high pass content of the texture is well captured by the adaptive dictionary, some low pass residual content is visible in the curvelet layer, mainly because elongated curvelets atoms are able to match some patterns of the texture.

We note that in both examples of Figures 5.1 and 5.2, the cartoon layer is not perfectly recovered by the adaptive MCA method. In particular, it contains residual oscillations of the original texture layer. This is a consequence of the mutual coherence between the fixed and learned dictionaries. Indeed, both wavelets and curvelets atoms are oscillating and thus capture some of the texture content. The quality of the cartoon layer can be enhanced by adding a total variation penalty as detailed in [55] to direct this component to better fit the piecewise-smooth model.



Original u_1 Original u_2 Mixture $f = u_1 + u_2 + \varepsilon$



Adaptive MCA $u_1^\star$ Adaptive MCA $u_2^\star$

FIG. 5.2. *Top row: original component to recover and observed mixture. Bottom row: separation using adaptive MCA with a curvelet dictionary and a learned dictionary.*

Separation of two textures using exemplars. In this separation scenario, we consider an observed image $f = u_1 + u_2 + \varepsilon$ of $N = 256 \times 256$ pixels, where each u_s corresponds to a stationary texture. We also consider two exemplar textures $(\tilde{u}_1, \tilde{u}_2)$ of 128×128 pixels that are similar to the components to retrieve. In practice, both u_s and $\tilde{u}_s$ are extracted from the same larger image, at different locations.

The learned dictionaries D_1, D_2 are optimized during the adaptive MCA Algorithm 1. They are initialized using the exemplars $(\tilde{u}_1, \tilde{u}_2)$ by minimizing (4.9) where U_s is composed of patches extracted from the exemplars.

Figure 4.1 shows the two exemplars together with the initialized dictionaries learned from these exemplars. Figure 5.3 shows the separation obtained with our adaptive MCA, which is of high visual quality, because the two textures exhibit a large morphological diversity.

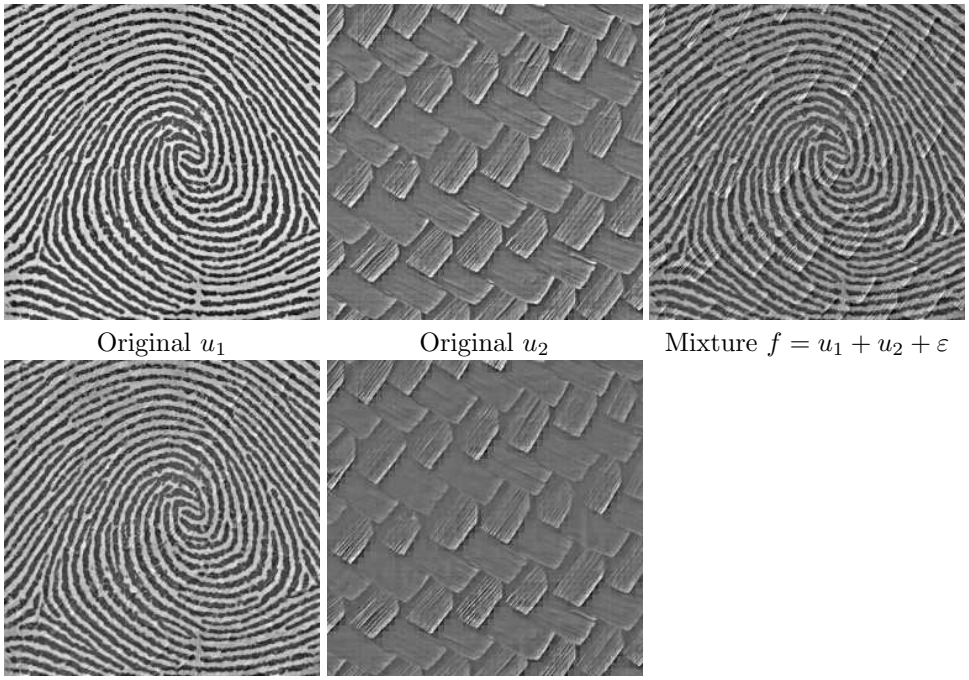


FIG. 5.3. *Example of texture separation using learned dictionaries initialized from exemplars.*

Separation of two textures with user intervention. To study the ability of the adaptive MCA to discriminate two complicated textures, Figure 5.4, left, shows a synthetic image obtained by linearly blending two textures $(\tilde{u}_1, \tilde{u}_2)$ as follow

$$f[i] = \gamma[i]\tilde{u}_1[i] + (1 - \gamma[i])\tilde{u}_2[i] + \varepsilon[i]$$

where γ is linearly decaying from 1 to 0 along the horizontal axis.

We use two adapted dictionaries (D_1, D_2) to perform the separation. The user intervention is required to initialize these dictionaries by extracting random patches respectively from the left and the right part of the observed image f.

Figures 5.5 and 5.6 show the application of the adaptive MCA to natural images, where no ground-truth (oracle) separation result is known. The user intervention is required to give an approximate location of the pattern to extract for each layer u_s.

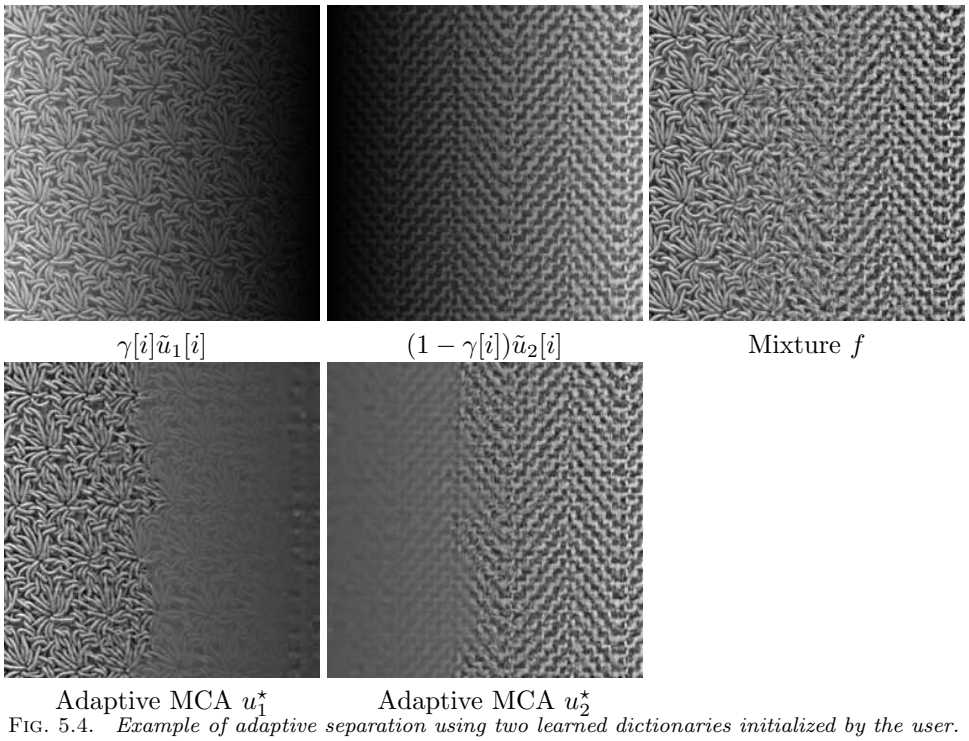


FIG. 5.4. *Example of adaptive separation using two learned dictionaries initialized by the user.*

FIG. 5.5. *Example of adaptive separation of a natural image into a cartoon layer and a texture layer. The rectangle shows the region given by the user to initialize the dictionary.*

Discussion about adaptive texture separation. We note that our method does not provide a fully unsupervised texture separation since the user needs to indicate roughly the locations where each textural pattern is present. In our approach, the separation process is not penalized if atoms are exchanged without due between different dictionaries, which explains the need for user intervention. A possible extension of our approach might be to include additional constraints $D_s \in \mathcal{D}_s$ beyond imposing zero-mean and bounded norm (4.3). For instance, one could force the atoms of a given dictionary to have some frequency localization, anisotropy or orientation by adding a set of convex constraints. Another extension of our method could integrate inter-dictionary penalties such as the discriminative learning detailed in [34].

5.2. Inpainting Small Gaps. Figure 5.7 depicts an example of inpainting to restore an image $f = \mathcal{K}u + \varepsilon$ where 65% of the pixels are missing. The original

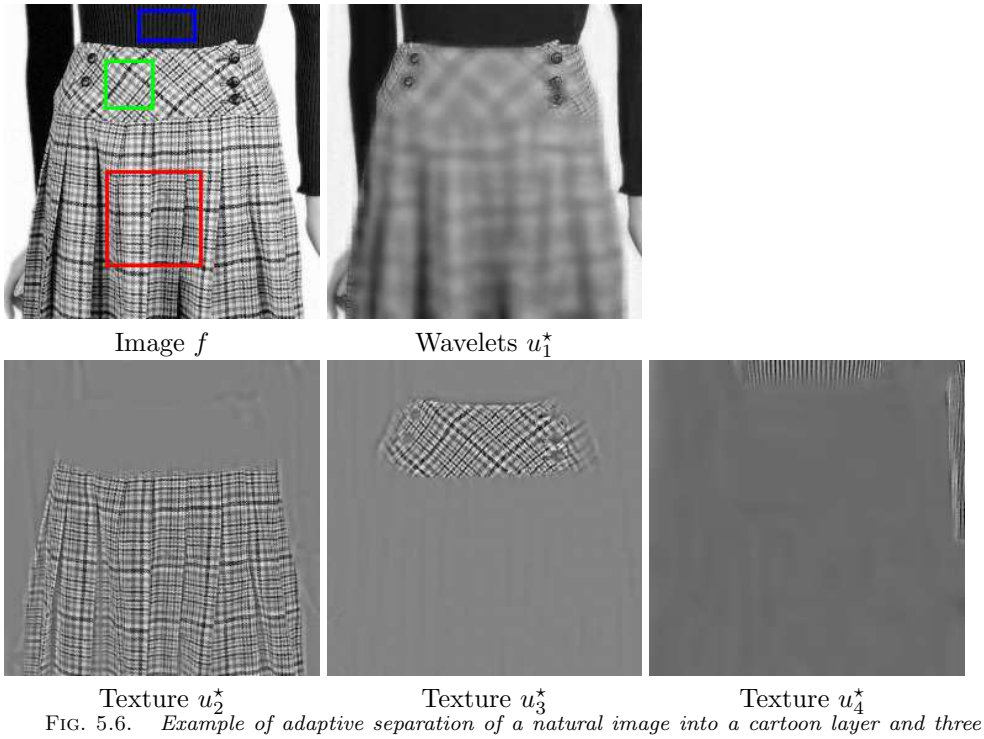


FIG. 5.6. *Example of adaptive separation of a natural image into a cartoon layer and three texture layers. The rectangles show the regions given by the user to initialize the dictionaries.*

image $u = \sum_{s=1}^{3} u_s$ is a superposition of a piecewise-smooth cartoon layer, a locally oscillating texture (scarf), and a texture similar to the one in Figure 5.1. This figure compares the result of inpainting with and without an additional layer corresponding to the learned dictionary D_3, with redundancy $m_3/n = 2$. In both results, the fixed dictionaries D_{wav} and D_{dct} were used. One can see that the inpainting result with a learned dictionary is able to recover the missing fine scale details of the complex texture, which is not the case with the use of a fixed local cosine dictionary D_{dct} alone to represent the texture.

Figure 5.8 shows another example of inpainting with gaps of medium size. Again, the inpainting with an additional learned dictionary D_3 brings some improvement over the inpainting obtained using standard MCA with only fixed dictionaries, although the improvement is visually less important than with the random mask used in Figure 5.7. Our method however not only solves the inpainting problem, but also yields an intuitive separation of the resulting inpainted image, which might be relevant for some applications. Such applications include object tracking and recognition [58, 56], edge detection after getting rid of texture [55], or textured artifacts removal in imaging devices such as astronomy [54].

Discussion about adaptive texture inpainting. We note that similarly to other dictionary learning approaches [36] and related methods such as fields of experts [51], our approach is restricted to the inpainting of small gaps. The sparsity regularization is indeed efficient if the width of missing regions is smaller that the width τ of the patches. For large gaps, these methods tend to produce a blurry reconstruction in the middle of the missing region. Inpainting large missing regions corresponds to a synthesis problem and should be attacked under the umbrella of other approaches based either on high order PDEs for cartoon inpainting [10, 5, 3], or patch recopy

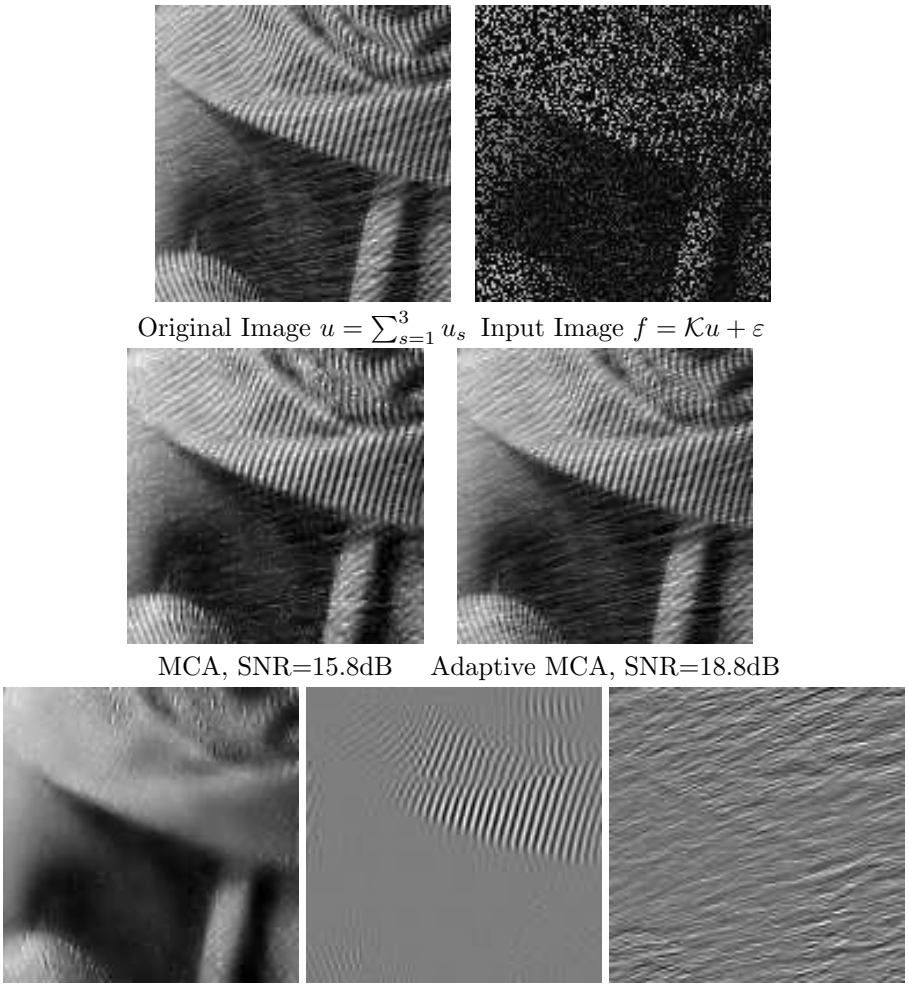


FIG. 5.7. *Top row: original image and masked image with 65% randomly removed pixels. Middle row: inpainted images provided by the standard and adaptive MCA algorithms. Bottom: the three layers provided by the adaptive MCA algorithm.*

[19, 13] for more general (texture) pattern inpainting.

Conclusion. We have proposed a new adaptive provably convergent algorithm to perform structure and texture separation using both fixed and learned dictionaries, along with an application to the simultaneous separation and inpainting problem. The main feature of the method is its ability to jointly decompose the image and learn the local dictionaries, which allows to adapt the process to the properties of the textures to be extracted. Numerical examples have shown that this adaptivity improves the efficiency and visual quality of the separation and inpainting. We have also shown that handling several learned dictionaries is possible, but this requires a special care to be taken at the initialization to achieve the desired separation effect.

REFERENCES

[1] M. AHARON, M. ELAD, AND A.M. BRUCKSTEIN, The K-SVD: An algorithm for designing of

Input Image MCA, SNR=24.9dB Adaptive MCA, SNR=28.6dB

FIG. 5.8. *Image inpainting with a structured mask containing moderately large gaps.*

overcomplete dictionaries for sparse representation, IEEE Trans. On Signal Processing, 54 (2006), pp. 4311–4322.

[2] J. F. AUJOL, G. AUBERT, L. BLANC-FERAUD, AND A. CHAMBOLLE, Image decomposition into a bounded variation component and an oscillating component, Journal of Math. Im. and Vision, 22 (2005), pp. 71–88.

[3] C. BALLESTER, M. BERTALMIO, V. CASELLES, G. SAPIRO, AND J. VERDERA, Filling-in by joint interpolation of vector fields and gray levels, IEEE Transaction on Image Processing, 10 (2001), pp. 1200–1211.

[4] A. J. BELL AND T. J. SEJNOWSKI, The independent components of natural scenes are edge filters, Vision Research, (1997), pp. 3327–3338.

[5] M. BERTALMÌO, G. SAPIRO, V. CASELLES, AND C. BALLESTER, Image inpainting, in Siggraph 2000, 2000, pp. 417–424.

[6] J. BIOUCAS-DIAS, Bayesian wavelet-based image deconvolution: a gem algorithm exploiting a class of heavy-tailed priors, IEEE Transactions on Image Processing, 15 (2006), pp. 937–951.

[7] J. BOBIN, J-L. STARCK, M.J. FADILI, AND Y. MOUDDEN, Sparsity and morphological diversity in blind source separation, IEEE Transactions on Image Processing, 16 (2007), pp. 2662 – 2674.

[8] J. BOBIN, J.-L STARCK, M. J. FADILI, Y. MOUDDEN, AND D. L. DONOHO, Morphological component analysis: An adaptive thresholding strategy, IEEE Transactions on Image Processing, 16 (2007), pp. 2675 – 2681.

[9] E. CANDÈS AND D. DONOHO, New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities, Comm. Pure Appl. Math., 57 (2004), pp. 219–266.

[10] V. CASELLES, J. M. MOREL, AND C. SBERT, An axiomatic approach to image interpolation, IEEE Transaction on Image Processing, 7 (1998), pp. 376–386.

[11] S. S. CHEN, D.L. DONOHO, AND M.A. SAUNDERS, Atomic decomposition by basis pursuit, SIAM Journal on Scientific Computing, 20 (1998), pp. 33–61.

[12] P. L. COMBETTES AND V. R. WAJS, Gsignal recovery by proximal forward-backward splitting, SIAM Journal on Multiscale Modeling and Simulation, 4 (2005).

[13] A. CRIMINISI, P. PÉREZ, AND K. TOYAMA, Region filling and object removal by exemplar-based image inpainting, IEEE Transactions on Image Processing, 13 (2004), pp. 1200–1212.

[14] I. DAUBECHIES, M. DEFRISE, AND C. DE MOL, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Comm. Pure Appl. Math, 57 (2004), pp. 1413–1541.

[15] L. DEMANET AND L. YING, Wave atoms and sparsity of oscillatory patterns, Applied and Computational Harmonic Analysis, 23 (2007), pp. 368–387.

[16] D. DONOHO, Wedgelets: Nearly-minimax estimation of edges, Ann. Statist, 27 (1999), pp. 353–382.

[17] D. DONOHO AND I. JOHNSTONE, Ideal spatial adaptation via wavelet shrinkage, Biometrika, 81 (1994), pp. 425–455.

[18] B. EFRON, T. HASTIE, I. JOHNSTONE, AND T. TIBSHIRANI, Least angle regression, Annals of Statistics, 32 (2004), p. 407?499.

[19] A. A. EFROS AND T. K. LEUNG, Texture synthesis by non-parametric sampling, in ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2, IEEE Computer Society, 1999, p. 1033.

[20] M. ELAD AND M. AHARON, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. on Image Processing, 15 (2006), pp. 3736–3745.

[21] M. Elad, P. Milanfar, and R. Rubinstein, Analysis versus synthesis in signal priors, Inverse Problems, 23 (2007), pp. 947–968.
[22] M. Elad, J.-L Starck, D. Donoho, and P. Querre, Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA), Applied and Computational Harmonic Analysis, 19 (2005), pp. 340–358.
[23] K. Engan, S. O. Aase, and J. Hakon Husoy, Method of optimal directions for frame design, in Proc. ICASSP '99, Washington, DC, USA, 1999, IEEE Computer Society, pp. 2443–2446.
[24] M.J. Fadili, J-L. Starck, and F. Murtagh, Inpainting and zooming using sparse representations, The Computer Journal, 52 (2007), pp. 64–79.
[25] M. J. Fadili and J.-L. Starck, Sparse representation-based image deconvolution by iterative thresholding, in Astronomical Data Analysis IV, F. Murtagh and J.-L. Starck, eds., Marseille, France, 2006.
[26] M. Figueiredo and R. Nowak, An EM algorithm for wavelet-based image restoration, IEEE Transactions on Image Processing, 12 (2003), pp. 906–916.
[27] ———, A bound optimization approach to wavelet-based image deconvolution, in IEEE ICIP, 2005.
[28] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T-W. Lee, and T.J. Sejnowski, Dictionary learning algorithms for sparse representation, Neural Comput., 15 (2003), pp. 349–396.
[29] E. Le Pennec and S. Mallat, Bandelet Image Approximation and Compression, SIAM Multiscale Modeling and Simulation, 4 (2005), pp. 992–1039.
[30] S Lesage, R Gribonval, F Bimbot, and L Benaroya, Learning unions of orthonormal bases with thresholded singular value decomposition, in IEEE ICASSP, 2005, pp. 293–296.
[31] M. S. Lewicki and T. J. Sejnowski, Learning overcomplete representations, Neural Comput., 12 (2000), pp. 337–365.
[32] L. Lieu and L. Vese, Image restoration and decompostion via bounded total variation and negative hilbert-sobolev spaces, 2005. UCLA CAM Report 05-33.
[33] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, Online dictionary learning for sparse coding, to appear in Journal of Machine Learning Research, (2010).
[34] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, Discriminative learned dictionaries for local image analysis, in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[35] ———, Supervised dictionary learning, Proc. Advances Neural Information Processing Systems, (2008).
[36] J. Mairal, M. Elad, and G. Sapiro, Sparse representation for color image restoration, IEEE Transaction on Image Processing, 17 (2008), pp. 53–69.
[37] S. Mallat, A Wavelet Tour of Signal Processing, Academic Press, San Diego, 1998.
[38] S. Mallat, Geometrical grouplets, to Appear in Applied and Computational Harmonic Analysis, (2008).
[39] S. Mallat and G. Peyré, Orthogonal bandlet bases for geometric images approximation, To appear in Com. Pure and Applied Mathematics, (2006).
[40] S. Masnou, Disocclusion: a variational approach using level lines, IEEE Transaction on Image Processing, 11 (2002), pp. 68–76.
[41] F. G. Meyer and R. R. Coifman, Brushlets: A tool for directional image analysis and image compression, Journal of Appl. and Comput. Harmonic Analysis, 5 (1997), pp. 147–187.
[42] Y. Meyer, Oscillating Patterns in Image Processing and Nonlinear Evolution Equations, American Mathematical Society, Boston, MA, USA, 2001.
[43] Y. Nesterov, Smooth minimization of non-smooth functions, Math. Program., 103 (2005), pp. 127–152.
[44] Yu. Nesterov, Gradient methods for minimizing composite objective function, CORE Discussion Papers 2007076, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), Sept. 2007.
[45] B. A. Olshausen and D. J. Field, Emergence of simple-cell receptive-field properties by learning a sparse code for natural images, Nature, 381 (1996), pp. 607–609.
[46] M. R. Osborne, Brett Presnell, and B. A. Turlach, A new approach to variable selection in least squares problems, IMA Journal of Numerical Analysis, 20 (2000), pp. 389–403.
[47] A. Ozerov and C. Févotte, Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation, IEEE Trans. Audio, Speech and Language Processing, (2010). to appear.
[48] G. Peyré, Sparse modeling of textures, Journal of Mathematical Imaging and Vision, 34 (2009), pp. 17–31.
[49] ———, Texture processing with grouplets, to appear in IEEE Trans. on Pattern Analysis and

Matching Intelligence, (2009).
[50] G. Peyré, J. Fadili, and J.L. Starck, Learning dictionaries for geometry and texture separation, in Proceedings of SPIE Wavelets XII, SPIE, 2007.
[51] S. Roth and M. J. Black, Fields of experts, International Journal of Computer Vision, 82 (2009), pp. 205–229.
[52] L. I. Rudin, S. Osher, and E. Fatemi, Nonlinear total variation based noise removal algorithms, Phys. D, 60 (1992), pp. 259–268.
[53] N. Shoham and M. Elad, Alternating ksvd-denoising for texture separation, in The IEEE 25-th Convention of Electrical and Electronics Engineers in Israel, IEEE Computer Society, 2008.
[54] J.-L. Starck, M. Elad, and D.L. Donoho, Redundant multiscale transforms and their application for morphological component analysis, Advances in Imaging and Electron Physics, 132 (2004).
[55] ———, Image decomposition via the combination of sparse representation and a variational approach, IEEE Transaction on Image Processing, 14 (2005), pp. 1570–1582.
[56] D. Tschumperlé, Y. Bentolila, J. Martinot, and M.J. Fadili, Fast time-space tracking of smoothly moving fine structures in image sequences., in IEEE ICIP, San Antonio, 2007.
[57] D. Tschumperlé and R. Deriche, Vector-valued image regularization with PDEs: Acommon framework for different applications, IEEE Trans. Pattern Anal. Mach. Intell, 27 (2005), pp. 506–517.
[58] D. Tschumperlé, M. J. Fadili, and Y. Bentolila, Wire structure pattern extraction and tracking from x-ray images of composite mechanisms, in IEEE CVPR'06, New-York, USA, 2006.
[59] P. Tseng, Convergence of a block coordinate descent method for nondifferentiable minimization, Journal of Optimization Theory and Applications, 109 (2001), pp. 475–494.
[60] L.A. Vese and S.J. Osher, Modeling textures with total variation minimization and oscillating patterns in image processing, Journal of Scientific Computing, 19 (2003), pp. 553–572.
[61] M. Zibulevsky and B. A. Pearlmutter, Blind source separation by sparse decomposition in a signal dictionary, Neural Computation, 13 (2001), pp. 863–882.